

## TP4 : Base de données Relationnelles sous Oracle : Langage PL/SQL (Les Procédures, Les Fonctions, les Exceptions et les Déclencheurs)

### Objectifs :

- Manipuler une base de données par des requêtes avancées en utilisant des programme PL/SQL.
- Automatiser le traitement des données par des déclencheurs.
- Savoir gérer les exceptions et les erreurs dans les programmes PL/SQL

**Travail à faire :** Dans les exercices qui suivent nous allons utiliser la même BDD du TP1 « BDDVente »

### Exo-1

- Créer une vue (CAANN) qui permet de calculer le Montant des ventes par année.
- Créer une fonction stockée qui permet de renvoyer le montant des ventes pour une année dont la valeur est transmise en argument. La fonction doit aussi gérer le cas où il n'y a pas de ventes réalisées pendant l'année choisie (Montant des ventes = 0).
- Créer un programme pour afficher les Montant des ventes pendant les années 2004...2018

| ANNEE | Nombre    |
|-------|-----------|
| 2004  | .00       |
| 2005  | .00       |
| 2006  | .00       |
| 2007  | 223731.11 |
| 2008  | 668085.52 |
| 2009  | 516528.40 |
| 2010  | .00       |

**Exo2 :** Nous voulons stocker dans une table de base de données les clients (Année, code, société, CA) qui ont réalisé le meilleur chiffre d'affaires entre deux année ANN1 et ANN2. Pour cela :

- Créer un package « Pack » contenant :

- Un type « Str » enregistrement pour stocker le id client, le nom de la société et le chiffre d'affaires du client
- Une fonction stockée prenant comme paramètre un entier qui représente une année et renvoie un enregistrement qui correspond au client qui a réalisé le meilleur chiffre d'affaires. la fonction doit renvoyer l'enregistrement (*valeur\_ANN, 0, aucunclient, 0*) s'il n y a pas de vente réalisée pendant l'année ANN (voir exemple).

**Function CA\_CL\_Max(ANN integer) Return Str;**

- Une procédure stockée qui a comme paramètre d'entrée deux années ANN1 et ANN2 et stocke dans une table de base de données 'Table\_Max\_CA\_ANN' les clients qui ont réalisé le plus grand CA entre les années ANN1 et ANN2

**Procedure CA\_CL(ANN1 integer, ANN2 integer);**

- 2- Créer un programme PLSQL qui appelle la procédure CA\_CL() pour les années [2014..2025]. Ci-dessous les données insérées dans une table de la BDD :

| ANNEE | CODECLIENT | NOMCLIENT                        | MT             |
|-------|------------|----------------------------------|----------------|
| 2014  | 867        | Accumsan Neque Et Ltd            | 175262,1464584 |
| 2015  | 914        | Aliquam Associates               | 172845,3813895 |
| 2016  | 723        | Ullamcorper Viverra LLC          | 183105,2637129 |
| 2017  | 409        | Fringilla Porttitor Incorporated | 179766,5586037 |
| 2018  | 189        | Enim Sed Foundation              | 178653,2870428 |
| 2019  | 222        | Eu Accumsan Limited              | 152329,614498  |
| 2020  | 595        | Ac Eleifend Vitae PC             | 163348,3162764 |
| 2021  | 0          | aucun client                     | 0              |
| 2022  | 0          | aucun client                     | 0              |
| 2023  | 0          | aucun client                     | 0              |
| 2024  | 0          | aucun client                     | 0              |
| 2025  | 0          | aucun client                     | 0              |

**Exo3 :** Même exercice que l'exo2 mais cette fois nous voulons stocker dans une table BDD les trois premiers clients qui ont réalisé les meilleurs CA (rownum<=3). Utiliser le Ref\_Cursor pour renvoyer les records de la fonction CA\_CL\_Max à la procédure CA\_CL.

**Function CA\_CL\_Max(ANN integer) Return sys\_refcursor.**

**Exo4 :** Créer un trigger qui permet de vérifier la disponibilité du produit avant chaque insertion d'une nouvelle ligne de commande dans une commande déjà créée. Ce trigger doit aussi faire la mise à jour du stock si le produit est disponible.

**Exo5:**

- 1- Créer une fonction stockée qui prend en paramètre un entier qui représente le numéro du mois et renvoi la saison correspondante : **Saison(Mo integer) Return Varchar2**

Les saisons en fonction des mois sont données par :

Hiver : 12 → 2

Printemps : 3 → 6

Été : 7 → 9

Automne : 10 → 11

- 2- Créer la vue CA\_Prod\_Saison qui permet de calculer le Montant des ventes par produit et par saison.
- 3- Créer une fonction stockée **MaxSaison(X varchar2) return CA\_Prod\_Saison%rowtype** qui permet de renvoyer le produit qui a réalisé le plus grand montant des ventes pendant la saison qui sera transmise en argument de cette fonction.

**Exo6 :**

Créer une Procédure stockée qui charge en mémoire, dans une collection T(), tous les clients avec pour chacun deux les commandes qui a réalisé. Cette collection sera indexée par le « idclient » : T(idclient). Cette procédure doit aussi afficher un client avec ces

commandes en fixant l'indice de la collection. Ce dernier sera un paramètre d'entrée de la procédure (voir figure). Ci-dessous un exemple d'exécution de cette procédure :

```
execute table_client(298);
```

```
298 aptent Molestie Tellus Aenean Incorporated
98 16/01/11 298
2290 21/10/14 298
1514 26/06/17 298
1720 23/04/13 298
2823 26/09/18 298
2861 18/12/13 298
4938 17/08/20 298
5005 04/12/12 298
3921 03/03/18 298
4345 05/10/13 298
6122 19/07/14 298
7793 02/12/10 298
6492 18/12/13 298
7875 21/05/19 298
9376 05/05/19 298
```

Procédure PL/SQL terminée.

